

## АННОТАЦИЯ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ

Дисциплина Б1.В.ОД.4 Введение в теорию алгоритмов и алгоритмические языки

**Семестр:** 4

**Количество часов:** 144

**Количество зачетных единиц:** 4

**Курсовая работа:** -

**Промежуточная аттестация:** дифференцированный зачет

### **Место дисциплины в структуре ООП:**

Дисциплина «Введение в теорию алгоритмов и алгоритмические языки» относится к вариативной части блока «Дисциплины (модули)» Б1.В.ОД.4 учебного плана подготовки бакалавра по направлению 09.03.03 *Прикладная информатика* направленность «Прикладная информатика в информационной сфере».

Дисциплина «Введение в теорию алгоритмов и алгоритмические языки» предполагает предварительное изучение обучающимися дисциплин: «Математика», «Теория вероятностей и математическая статистика», «Дискретная математика», «Теория систем и системный анализ», «Физика», «Экономико-математические модели и методы», «Компьютерная оптимизация», «Алгебра и геометрия», «Информатика и программирование», «Интеллектуальные информационные системы», дополняет дисциплины: «Математическая логика», «Компьютерное решение задач имитационным методом», «Вычислительная математика».

Освоение дисциплины необходимо как предшествующее при изучении следующих дисциплин: «Программирование», «Программирование в среде 1С: Предприятие», «Программирование в среде 1С: Бухгалтерия», «Программная инженерия», «Автоматизированные бухгалтерские информационные системы», «Инженерия знаний».

**Цель дисциплины:** освоение базовых понятий и терминов программирования как науки.

### **Задачи:**

- освоение основных алгоритмов обработки данных;
- ознакомление обучающихся с конструкциями языка программирования высокого уровня и технологией разработки программ на языке высокого уровня;
- изучение основных структур данных и алгоритмов их обработки;
- изучение базовых концепций парадигм объектно-ориентированного и параллельного программирования.

### **Содержание дисциплины:**

Общее понятие алгоритма и краткий обзор существующих алгоритми-

ческих языков. Неформальный алгоритмический язык – псевдокод, максимально приближенный к естественному языку. Основные конструкции алгоритмического языка – алгоритм, ветвление, цикл; простейшие примеры программ на псевдокоде. Понятие переменной.

Алфавит, синтаксис и семантика языка программирования. Средства определения синтаксиса: расширенные формулы Бэкуса-Наура (РБНФ), синтаксические диаграммы. Классификация языков программирования по уровню абстракции. Обзор основных элементов языка программирования высокого уровня.

Интуитивное определение алгоритма. Основные свойства алгоритмов. Понятие исполнителя алгоритма. Основные методы разработки алгоритмов. Формальное определение алгоритма. Машина Тьюринга. Операции над машинами Тьюринга. Тезис Тьюринга. Нормальные алгоритмы Маркова. Принцип нормализации Маркова. Понятия самоприменимости алгоритма и алгоритмической неразрешимости.

Рекурсивные алгоритмы. Понятие рекурсивного алгоритма. Виды рекурсии. Структура рекурсивного алгоритма. Примеры рекурсивных алгоритмов. Реализация перебора с возвратом с помощью рекурсии. Реализация механизма рекурсии. Сравнение рекурсии и итерации. Рекурсивные данные. Бинарные деревья. Обходы бинарного дерева. Основные алгоритмы обработки бинарных деревьев (уничтожение дерева, вставки информации в дерево, удаление вершины из дерева, сравнения двух деревьев и др.).

Понятие абстрактного типа данных. Классификация структур данных. Последовательные списки: стек, очередь, дек. Связные списки: однонаправленный список, двунаправленный список, циклический список. Графы. Методы реализации графов: матрица смежности, матрица инцидентности, списки смежных вершин, список ребер.

Внутренняя сортировка (сортировка массивов). Понятие сложности алгоритма сортировки. Основные алгоритмы внутренней сортировки: пузырьковая сортировка, шейкерная сортировка, сортировка простым выбором, сортировка простыми вставками, сортировка Шелла, сортировка слиянием, быстрая сортировка. Внешняя сортировка (сортировка файлов). Отрезки файла. Операции разделения файла и слияния файлов. Некоторые алгоритмы внешней сортировки: многофазная сортировка, сбалансированное слияние. Использование алгоритмов внутренней сортировки в сортировке последовательных файлов.

Поиск в массиве. Линейный поиск. Бинарный поиск. Поиск в таблице. Хеширование. Выбор хеш-функции. Разрешение коллизий: метод открытой адресации, метод цепочек.

Парадигма объектно-ориентированного программирования (ООП): концепции объекта и класса, инкапсуляции, наследования и полиморфизма. Спецификация видимости атрибутов и методов объекта. Механизмы раннего и позднего связывания. Статические и виртуальные методы. Иерархии классов. Шаблоны классов.

Событийно-управляемое программирование. Пользовательские и сис-

темные события в программе. Методы обработки и распространение событий. Управление параллелизмом с помощью механизма обработки событий. Прикладной программный интерфейс (Application Programming Interface), API-программирование. Методы обработки данных, основанные на компонентных технологиях. Промежуточное программное обеспечение (middleware).

Параллельная вычислительная система. Примеры больших задач. Режимы выполнения задач: последовательный, псевдопараллельный (разделение времени) и параллельный. Виды параллелизма: многопроцессорная, векторная и конвейерная обработка. Методика разработки параллельных алгоритмов. Модель «процессы-каналы» параллельной программы. Разделение вычислений на независимые части: параллелизм по данным и функциональный параллелизм. Выделение информационных зависимостей: локальные и глобальные, статические и динамические схемы передачи данных, структурные и произвольные, синхронные и асинхронные способы взаимодействия. Масштабирование подзадач. Распределение подзадач по процессорам вычислительной системы.

**В результате освоения дисциплины** обучающийся должен обладать следующими компетенциями:

ОПК-3: способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности (*знать* основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности; *уметь* анализировать социально-экономические задачи и процессы с применением базовых алгоритмов; *владеть навыками* использования основных законов естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности);

ПК-8: способностью программировать приложения и создавать программные прототипы решения прикладных задач (*знать* базовые алгоритмы для работы с информацией; *уметь* строить алгоритмы решения прикладных задач и оценивать их сложность; *владеть навыками* программирования и тестирования программ).

### **Образовательные технологии:**

В преподавании дисциплины «Введение в теорию алгоритмов и алгоритмические языки» применяются разнообразные интерактивные образовательные технологии в зависимости от вида и цели учебного занятия.

Теоретический материал излагается на лекционных занятиях в форме проблемно-ориентированных лекций.

Лабораторные занятия по дисциплине «Введение в теорию алгоритмов и алгоритмические языки» ориентированы на закрепление теоретического материала, изложенного на лекционных занятиях, а также на приобретение

дополнительных знаний, умений и практических навыков осуществления аналитической и профессиональной деятельности с применением интерактивных форм обучения (моделирования деловых ситуаций, подготовка презентаций, мастер-классы и др.).

**Составитель:** А. Г. Калинин, канд. техн. наук, доцент, кафедра информатики и естественнонаучных дисциплин.